

CLAIMS

We claim:

1. A method for detecting whether a routine has stalled, comprising the steps of:

5       accessing existing code for a first routine;  
automatically modifying said existing code to include new code; and  
using said new code to determine if said first routine has stalled.

10       2. A method according to claim 1, wherein:  
said existing code is object code; and  
said new code is object code.

15       3. A method according to claim 2, further comprising the step of:  
receiving a rule, said rule identifies said first routine and an interval, said step  
of automatically modifying is performed in response to said rule, said first routine is  
considered to be stalled after a determination that said first routine has been running  
for at least as long as said interval.

20       4. A method according to claim 1, wherein said step of automatically  
modifying comprises the steps of:

adding code for a timing mechanism to said existing code;  
adding code for starting said timing mechanism to said existing code;  
adding code for stopping said timing mechanism to said existing code;  
adding a first instruction to said first routine, said first instruction causes the  
25       execution of said code for starting said timing mechanism; and  
adding a second instruction to said first routine, said second instruction  
causes the execution of said code for stopping said timing mechanism.

5. A method according to claim 4, wherein:

said second instruction is added such that it is executed at all exits of said routine.

6. A method according to claim 4, wherein:  
said first routine is a method.

7. A method according to claim 4, wherein:  
said first routine is a thread.

8. A method according to claim 4, wherein:  
said first routine is one of a plurality of routines that comprise a process.

9. A method according to claim 1, wherein said step of using comprises the steps of:

receiving an indication that said first routine has started;

starting a timing mechanism in response to said step of receiving;

receiving an indication that said first routine has completed, if said first routine has completed;

stopping said timing mechanism in response to receiving said indication that said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

10. A method according to claim 9, wherein said step of using further comprises the steps of:

accessing a current time;

verifying that said first routine is not known to be stalled or completed;

accessing said due time; and

determining whether first due time is earlier than said current time, said timing mechanism is overdue if said step of determining concludes that said first due time is earlier than said current time.

5           11.     A method according to claim 9, wherein said step of stopping said timing mechanism comprises the steps of:

          determining whether said first routine has been reported as being stalled;

          changing said reporting to no longer indicate that said first routine is stalled if  
10       said step of determining concludes that said first routine has been reported as being stalled; and

          stopping said timing mechanism if said first routine has not been reported as being stalled.

          12.     A method according to claim 9, wherein:

15       said first routine is a thread executing a method, said indication that said first routine has completed indicates that said thread has exited said method.

          13.     A method for detecting whether a routine has stalled, comprising the steps of:

20       receiving an indication that a particular routine is running, said particular routine is one of a plurality of routines that comprise a process; and  
          automatically determining whether said particular routine has stalled.

          14.     A method according to claim 13, wherein:

25       said particular routine is a thread.

          15.     A method according to claim 13, wherein:

          said plurality of routines are threads;

30       at least two or more of said threads, including said particular routine, are run concurrently.

16. A method according to claim 13, wherein:

said particular routine is a method

5

17. A method according to claim 16, further comprising the step of:

receiving an indication of said method, said step of determining whether said particular routine is stalled includes detecting a situation when a thread enters said method and does not return within an approximation of an expected time frame.

10

18. A method according to claim 13, wherein:

said indication that a particular routine is running is an indication that said particular routine has started; and

said step of determining whether said particular routine has stalled comprises the steps of:

15

starting a timing mechanism in response to said step of receiving an indication that a particular routine is running,

receiving an indication that said particular routine has completed, if said particular routine has completed,

20

stopping said timing mechanism in response to receiving said indication that said particular routine has completed, and

reporting said particular routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

19. A method according to claim 18, wherein:

25

said indication that said particular routine has started is received from within said particular routine; and

said indication that said particular routine has completed is received from within said particular routine.

30

20. A method according to claim 18, further comprising the step of:

automatically adding new code to existing code for said particular routine, said new code performs said step of starting a timing mechanism, stopping said timing mechanism and reporting.

5           21.     A method according to claim 18, further comprising the step of:  
automatically modifying existing object code for said particular routine in order to add new object code to said existing object code for said particular routine, said new object code performs said step of starting a timing mechanism, stopping said timing mechanism and reporting.

10           22.     A method according to claim 18, wherein said step of determining whether said particular routine has started further comprises the steps of:  
accessing a current time;  
verifying that said particular routine is not known to be stalled or completed;  
15           accessing a due time; and  
determining whether due time is earlier than said current time, said timing mechanism is overdue if it is determined that said due time is earlier than said current time.

20           23.     A method according to claim 18, wherein said step of stopping said timing mechanism comprises the steps of:  
determining whether said particular routine has been reported as being stalled;  
changing said reporting to no longer indicate that said particular routine is  
25           stalled if it is determined that said particular routine has been reported as being stalled; and  
stopping said timing mechanism if said particular routine has not been reported as being stalled.

24. A method according to claim 18, wherein said step of starting said timing mechanism comprises the steps of:

receiving a threshold;

accessing a current time;

5 determining a first due time based on said threshold and said current time; and

adding a indication of said particular routine and said first due time to a set of due times for other routines, said timing mechanism is overdue after said timing mechanism determines that said due time has been exceeded.

10

25. A method according to claim 24, wherein said step of stopping said timing mechanism comprises the steps of:

determining whether said particular routine has been reported as being stalled;

15

changing said reporting to no longer indicate that said particular routine is stalled if said particular routine has been reported as being stalled; and

removing said indication of said particular routine and said due time from said set of items if said particular routine has not been reported as being stalled.

20

26. A method according to claim 25, further comprising the step of:

repeatedly evaluating said set of due times to determine if any of said due times have passed.

25

27. A method according to claim 26, wherein said step of repeatedly evaluating comprises the steps of:

accessing a current time;

verifying that said particular routine is not known to be stalled or completed;

accessing said first due time; and

determining whether first due time is earlier than said current time, said timing mechanism is overdue if said step of determining concludes that said first due time is earlier than said current time.

5           28.     A method according to claim 27, further comprising the step of:  
automatically modifying existing object code for said particular routine in  
order to add new object code to said existing object code for said particular routine,  
said new object code performs said steps of starting a timing mechanism, stopping  
said timing mechanism and reporting.

10           29.     A method according to claim 13, further comprising the step of:  
reporting said particular routine as being stalled if said particular routine was  
determined to be stalled.

15           30.     A method according to claim 29, wherein:  
said particular routine is an instance of a defined routine; and  
said step of reporting includes incrementing a counter that represents a  
number of instances of said defined routine that are currently stalled and reporting  
said number of instances of said defined routine that are currently stalled.

20           31.     A method according to claim 29, wherein:  
said particular routine is an instance of a defined routine; and  
said step of reporting includes determining and reporting how many instances  
of said defined routine were stalled at a specified time.

25           32.     A method according to claim 29, wherein:  
said particular routine is an instance of a defined routine; and  
said step of reporting includes receiving a customizable specified time period  
and reporting how many instances of said defined routine were stalled during said  
30     specified time period.

33. A method for detecting whether a thread has stalled, comprising the steps of:

receiving an indication that a particular thread is running; and  
determining whether said particular thread has stalled.

34. A method according to claim 33, further comprising the step of:

receiving an indication of a first method, said step of determining whether said particular thread is stalled includes detecting a situation when said thread enters said first method and does not return within an approximation of an expected time frame.

35. A method according to claim 33, further comprising the step of:

automatically modifying existing object code in order to add new object code, said new object code performs said step of determining.

36. A method according to claim 33, wherein:

said indication that a particular thread is running is an indication that said particular thread has started; and

said step of determining whether said particular thread has stalled comprises the steps of:

starting a timing mechanism in response to said step of receiving an indication that said particular thread has started,

receiving an indication that said particular thread has completed, if said particular thread has completed,

stopping said timing mechanism in response to receiving said indication that said particular thread has completed, and

reporting said particular thread as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

37. A method according to claim 36, wherein said step of stopping said timing mechanism comprises the steps of:

determining whether said particular routine has been reported as being stalled;

5 changing said reporting to no longer indicate that said particular routine is stalled if it is determined that said particular routine has been reported as being stalled; and

stopping said timing mechanism if said particular routine has not been reported as being stalled.

10

38. A method according to claim 33, wherein:

said particular thread is one of multiple threads running concurrently and which comprise a process.

15

39. A method for detecting whether a method has stalled, comprising the steps of:

receiving an indication that a particular method is running; and  
determining whether said method thread has stalled.

20

40. A method according to claim 39, further comprising the step of:

receiving an indication that identifies said particular method from a set of methods, said step of determining whether said particular method is stalled includes detecting a situation when a thread enters said particular method and does not return within an approximation of an expected time frame.

25

41. A method according to claim 39, further comprising the step of:

automatically modifying existing object code for said particular method in order to add new object code, said new object code performs said step of determining.

30

42. A method according to claim 39 wherein:

said indication that a particular method is running is an indication that said particular method has started; and

said step of determining whether said particular method has stalled comprises the steps of:

starting a timing mechanism in response to said step of receiving an indication that a particular method is running,

receiving an indication that said particular method has completed, if said particular method has completed,

stopping said timing mechanism in response to receiving said indication that said particular method has completed, and

reporting said particular method as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

43. A method according to claim 42, wherein said step of stopping said timing mechanism comprises the steps of:

determining whether said particular method has been reported as being stalled;

changing said reporting to no longer indicate that said particular method is stalled if it is determined that said particular method has been reported as being stalled; and

stopping said timing mechanism if said particular method has not been reported as being stalled.

44. A method according to claim 39, wherein

said particular method is one of multiple methods running concurrently and which comprise a process.

45. A method for detecting whether a routine has stalled, comprising the steps of:

receiving an indication that a first routine has started;

starting a timing mechanism in response to said indication that said first routine has started;

receiving an indication that said first routine has completed, if said first routine has completed;

stopping said timing mechanism in response to receiving said indication that said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

46. A method according to claim 45, further comprising the step of:

automatically adding new code to existing code for said first routine, said new code performs said step of starting a timing mechanism, stopping said timing mechanism and reporting.

47. A method according to claim 45, further comprising the step of:

automatically modifying existing object code for said first routine in order to add new object code to said existing object code for said first routine, said new object code performs said steps of starting a timing mechanism, stopping said timing mechanism and reporting.

48. A method according to claim 45, further comprising the steps of:

accessing a current time;

verifying that said first routine is not known to be stalled or completed;

accessing a due time for said first routine; and

determining whether said due time is earlier than said current time, said timing mechanism is overdue if said step of determining concludes that said first due time is earlier than said current time.

49. A method according to claim 45, wherein said step of stopping said timing mechanism comprises the steps of:

determining whether said first routine has been reported as being stalled;

changing said reporting to no longer indicate that said first routine is stalled if said first routine has been reported as being stalled; and

stopping said timing mechanism if said first routine has not been reported as being stalled.

50. A method according to claim 45, wherein:

said first routine is an instance of a defined routine; and

said step of reporting includes incrementing a counter that represents a number of instances of said defined routine that are stalled and reporting said number of instances of said defined routine that are stalled.

51. A method according to claim 45, wherein:

said first routine is an instance of a defined routine;

said step of reporting includes receiving a customizable specified time period and reporting how many instances of said defined were stalled during said specified time period.

52. One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

accessing existing code for a first routine;

automatically modifying said existing code to include new code; and

using said new code to determine if said first routine has stalled.

53. One or more processor readable storage devices according to claim 52, wherein:

said existing code is object code; and

said new code is object code.

54. One or more processor readable storage devices according to claim  
5 53, wherein said method further comprises the step of:

receiving a rule, said rule identifies said first routine and an interval, said step  
of automatically modifying is performed in response to said rule, said first routine is  
considered to be stalled after a determination that said first routine has been running  
for at least as long as said interval.

55. One or more processor readable storage devices according to claim  
52, wherein:

said first routine is one of a plurality of routines that comprise a process.

56. One or more processor readable storage devices according to claim  
52, wherein said step of using comprises the steps of:

receiving an indication that said first routine has started;

starting a timing mechanism in response to said step of receiving;

receiving an indication that said first routine has completed, if said first  
20 routine has completed;

stopping said timing mechanism in response to receiving said indication that  
said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not  
stopped prior to a determination that said timing mechanism is overdue.

57. One or more processor readable storage devices according to claim  
52, wherein:

said first routine is a thread performing a method; and

said step of using includes determining whether said thread entered said  
30 method and did not return within a predetermined time period.

58. An apparatus, comprising:

one or more storage devices; and

5 one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising the steps of:

accessing existing code for a first routine,

automatically modifying said existing code to include new code, and

using said new code to determine if said first routine has stalled.

10 59. An apparatus according to claim 58, wherein:

said existing code is object code; and

said new code is object code.

15 60. An apparatus according to claim 59, wherein said method further comprises the step of:

receiving a rule, said rule identifies said first routine and an interval, said step of automatically modifying is performed in response to said rule, said first routine is considered to be stalled after a determination that said first routine has been running  
20 for at least as long as said interval.

61. An apparatus according to claim 58, wherein:

said first routine is one of a plurality of routines that comprise a process.

25 62. An apparatus according to claim 58, wherein said step of using comprises the steps of:

receiving an indication that said first routine has started;

starting a timing mechanism in response to said step of receiving;

30 receiving an indication that said first routine has completed, if said first routine has completed;

stopping said timing mechanism in response to receiving said indication that said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

5

63. An apparatus according to claim 58, wherein:

said first routine is a thread performing a method; and

said step of using includes determining whether said thread entered said method and did not return within a predetermined time period.

10

64. One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

15

receiving an indication that a particular routine is running, said particular routine is one of a plurality of routines that comprise a process; and

automatically determining whether said particular routine has stalled.

20

65. One or more processor readable storage devices according to claim 64, wherein:

said particular routine is a thread.

66. One or more processor readable storage devices according to claim 64, wherein:

25

said particular routine is a method.

67. One or more processor readable storage devices according to claim 66, further comprising the step of:

receiving an indication of said method, said step of determining whether said particular routine is stalled includes detecting a situation when a thread enters said method and does not return within an approximation of an expected time frame.

5           68.    One or more processor readable storage devices according to claim 64, wherein:

          said indication that a particular routine is running is an indication that said particular routine has started; and

          said step of determining whether said particular routine has stalled comprises  
10       the steps of:

          starting a timing mechanism in response to said step of receiving an indication that a particular routine is running,

          receiving an indication that said particular routine has completed, if said particular routine has completed,

15       stopping said timing mechanism in response to receiving said indication that said particular routine has completed, and

          reporting said particular routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

20           69.    One or more processor readable storage devices according to claim 68, wherein said method further comprises the step of:

          automatically modifying existing object code for said particular routine in order to add new object code to said existing object code for said particular routine, said new object code performs said step of starting a timing mechanism, stopping  
25       said timing mechanism and reporting.

          70.    One or more processor readable storage devices according to claim 68, wherein said step of stopping said timing mechanism comprises the steps of:

          determining whether said particular routine has been reported as being  
30       stalled;

changing said reporting to no longer indicate that said particular routine is stalled if it is determined that said particular routine has been reported as being stalled; and

5        stopping said timing mechanism if said particular routine has not been reported as being stalled.

71.     One or more processor readable storage devices according to claim 64, wherein:

      said particular routine is an instance of a defined routine; and

10        said step of automatically determining includes incrementing a counter that represents a number of instances of said defined routine that are currently stalled and reporting said number of instances of said defined routine that are currently stalled.

72.     An apparatus, comprising:

15        one or more storage devices; and

      one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising the steps of:

20        receiving an indication that a particular routine is running, said particular routine is one of a plurality of routines that comprise a process, and automatically determining whether said particular routine has stalled.

73.     An apparatus according to claim 72, wherein:

      said particular routine is a thread.

25        74.     An apparatus according to claim 72, wherein:

      said particular routine is a method.

75.     An apparatus according to claim 74, further comprising the step of:

receiving an indication of said method, said step of determining whether said particular routine is stalled includes detecting a situation when a thread enters said method and does not return within an approximation of an expected time frame.

5           76.     An apparatus according to claim 72, wherein:  
              said indication that a particular routine is running is an indication that said  
              particular routine has started; and  
              said step of determining whether said particular routine has stalled comprises  
              the steps of:  
10               starting a timing mechanism in response to said step of receiving an  
                  indication that a particular routine is running,  
                  receiving an indication that said particular routine has completed, if  
                  said particular routine has completed,  
                  stopping said timing mechanism in response to receiving said  
15               indication that said particular routine has completed, and  
                  reporting said particular routine as stalled if said timing mechanism is  
                  not stopped prior to a determination that said timing mechanism is overdue.

20           77.     An apparatus according to claim 72, wherein said method further  
              comprises the step of:  
              automatically modifying existing object code for said particular routine in  
              order to add new object code to said existing object code for said particular routine,  
              said new object code performs said step of starting a timing mechanism, stopping  
              said timing mechanism and reporting.

25           78.     An apparatus according to claim 72, wherein said step of stopping  
              said timing mechanism comprises the steps of:  
              determining whether said particular routine has been reported as being  
              stalled;

changing said reporting to no longer indicate that said particular routine is stalled if it is determined that said particular routine has been reported as being stalled; and

5           stopping said timing mechanism if said particular routine has not been reported as being stalled.

79.     An apparatus according to claim 72, wherein:

10           said particular routine is an instance of a defined routine; and  
said step of automatically determining includes incrementing a counter that represents a number of instances of said defined routine that are currently stalled and reporting said number of instances of said defined routine that are currently stalled.